Block sparse linear models for learning structured dynamical systems in aeronautics

Cédric Rommel, Joseph Frédéric Bonnans, Baptiste Gregorutti, Pierre Martinon

Block sparse linear models for learning structured dynamical systems in aeronautics

C. Rommel[1,2,3], J. F. Bonnans[1,2], B. Gregorutti[3] and P. Martinon[1,2]
INRIA[1], CMAP[2], Safety Line[3]
cedric.rommel@safety-line.fr · frederic.bonnans@inria.fr
baptiste.gregorutti@safety-line.fr · pierre.martinon@inria.fr

Abstract

This paper addresses an aircraft dynamical system identification problem, with the goal of using the learned models for trajectory optimization purposes. Our approach is based on multi-task regression. We present in this setting a new class of estimators that we call Block sparse Lasso, which conserves a certain structure between the tasks and some groups of variables, while promoting sparsity within these groups. An implementation leading to consistent feature selection is suggested, allowing to obtain accurate models, which are suitable for trajectory optimization. An additional regularizer is also proposed to help in recovering hidden representations of the initial dynamical system. We illustrate our method with numerical results based on real flight data from 25 medium haul aircraft, totaling 8 millions observations.

1 INTRODUCTION

1.1 Dynamical system identification problem

We consider in this paper the task of learning the state function g defining the behavior of a dynamical system:

$$\dot{\boldsymbol{x}} = g(t, \boldsymbol{x}, \boldsymbol{u}), \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{X} \subset \mathbb{R}^{d_x}$ and $\boldsymbol{u}(t) \in \mathbb{U} \subset \mathbb{R}^{d_u}$ denote respectively the state and control variables, and t denotes the time. We are mainly focused here on the dynamics of an aircraft during its climb phase.

Aircraft dynamics identification is essential in several engineering application, including for the optimization of flight trajectories, which allows a more efficient use of an airline fleet regarding, for example, CO_2 emissions and fuel consumption. In this context, common state and control variables are $\boldsymbol{x} = (h, V, \gamma, m)$ and $\boldsymbol{u} = (\alpha, N_1)$ (see table 1 for notations) and a standard model for the state equation is the following, where T, D, L, I_{sp} are supposed to

depend on a subset of $(\boldsymbol{x}, \boldsymbol{u})$:

$$\begin{cases} \dot{h} = & V \sin\gamma, & (2) \\ \dot{V} = & \dfrac{T(\boldsymbol{x},\boldsymbol{u})\cos\alpha - D(\boldsymbol{x},\boldsymbol{u}) - mg\sin\gamma}{m}, & (3) \\ \dot{\gamma} = & \dfrac{T(\boldsymbol{x},\boldsymbol{u})\sin\alpha + L(\boldsymbol{x},\boldsymbol{u}) - mg\cos\gamma}{mV}, & (4) \\ \dot{m} = & -\dfrac{T(\boldsymbol{x},\boldsymbol{u})}{I_{sp}(\boldsymbol{x},\boldsymbol{u})}. & (5) \end{cases}$$

Table 1: Variables nomenclature

Notation	**Meaning**
h	Aircraft altitude
V	Aircraft true airspeed (TAS)
γ	Path angle
m	Aircraft mass
α	Angle of attack (AOA)
N_1	Engines turbofan speed
T	Total thrust force
D, L	Drag and lift forces
I_{sp}	Specific impulse
ρ	Air density
M	Aircraft Mach number
SAT	Static air temperature
β	Pitch angle
C	Fuel flow

Despite this specific use case, which is central in our work, we believe that the techniques and framework presented hereafter could be used in many other engineering and scientific contexts where sets of differential equations need to be identified based on data.

In the application case of trajectory optimization, the model used for prediction is required to be fast to evaluate and differentiable. More particularly for aircraft trajectories, scalability is also a requirement, as lots of data are available for airlines which record thousands of variables every second of every commercial flight. We also restrict our scope to models which are easily interpretable by domain experts, which includes allowing to compute T, D, L and I_{sp}. This means that learning the function g from (1) summarizes here into using equations (2)-(5) to infer the hidden functions T, D, L and I_{sp}, based uniquely on observations of $\boldsymbol{u}, \boldsymbol{x}$ and $\dot{\boldsymbol{x}}$.

We propose in this paper a method addressing all the requirements cited above, based on a structured version of multi-task sparse linear regression.

1.2 Related work

Dynamical system identification According to the literature [Jategaonkar, 2006, Maine and Iliff, 1986, Klein and Morelli, 2006], two widely used approaches for

learning aircraft state equations are the Output-Error Method (OEM) and Filter-Error Method (FEM). Both are said to be dynamic identification techniques, in the sense that they evaluate the goodness of a certain candidate solution $\hat{g}$ by integrating it given some fixed controls and comparing the states obtained to the measured states (or some other variables measured). The difference between them lies in the uncertainty assumed for the state equation integration, a Kalman filter being used in the latter to estimate the states instead of a deterministic integration scheme. Recent advances include using neural networks for the state estimation part [Peyada and Ghosh, 2009].

The older Equation-Error Method (EEM) [Maine and Iliff, 1985] differs from the former by trying to fit the state derivative information from the dynamic system instead of matching the states themselves. This makes this class of method more scalable and avoids several pratical issues related to the nested estimation problem of integrating the states of a possibly nonlinear dynamical system. Drawbacks of this method include needing access to state derivatives data (which are usually obtained by numerical differentiation of noisy time series) [Morelli, 2006] and lower accuracy than OEM and FEM in terms of predicted state variables [Peyada et al., 2008]. Furthermore, in the presence of noisy and highly correlated regressors, as it is our case, it has been demonstrated in [Morelli, 2006] that standard EEM produces biased estimations of the model parameters. Frequency domain variations of EEM have since been proposed to mitigate this phenomenon.

Multi-task learning Since the introduction of the multi-task learning concept, benefits have been evidenced by the statistics and machine learning communities, including in particular when a common representation is assumed to be shared by all tasks [Caruana, 1997, Evgeniou et al., 2005]. It has been addressed briefly by the aircraft dynamics identification community for EEM approaches, with the different tasks corresponding to different manoeuvers [Morelli, 2006] or to different dynamic equations [Rommel et al., 2017].

Feature selection based on L^1 regularizer The motivation for sparse linear models in aircraft dynamics identification problems has been presented in [Rommel et al., 2017]. Since the introduction of the Lasso estimator by [Tibshirani, 1994], many variations have been proposed to perform L^1-norm based feature selection in several settings. For example, the extensively studied Group Lasso [Yuan and Lin, 2005] performs selection among groups of variables by making use of a L^1-regularization over the L^2-norms of subsets of features. Similarly, multi-task feature selection techniques, such as [Argyriou et al., 2008, Obozinski et al., 2006], make use of a mixed $L^{2,1}$-norm to find common representations for all tasks. As for some applications sparsity within the groups is desirable, additionally to the selection of groups of variables, the Sparse-group Lasso was proposed by [Friedman et al., 2010].

Moreover, in some contexts, as for example when regressors are strongly correlated, the Lasso selection is known to be inconsistent. This translates in practice by the selection of irrelevant variables, which can be evidenced by obtaining different sparsity patterns when solving the Lasso several times with small changes in the training set. The Adaptive Lasso, proposed in [Zou, 2006], was proved to carry consistent feature selection in this context by making use of

data-dependent weights added to the L^1 penalty. Nonetheless, this last approach needs in practice the tuning of at least an additional hyperparameter, when compared to the Lasso. Another consistent feature selection algorithm is the Bolasso [Bach, 2008], which is based on the idea of intersecting the sparsity patterns returned by the Lasso run on bootstrap replications of the training set.

1.3 Core contributions

Since OEM and FEM classes of approaches do not scale very well, we propose a variation of EEM delivering a model with all the desired properties cited in section 1.1. This is achieved by casting the system (2)-(5) into a multi-task regression problem, which is the subject of section 2. Such formulation ensures that the structure of the differential system is conserved and benefits from the tight coupling existing between equations. This idea is similar to [Rommel et al., 2017], with the difference that the multi-task regression model proposed herein is linear. Despite this difference, we also assume a sparse polynomial structure for the functions T, D, L and I_{sp}. This not only complies with information found in the flight mechanics literature, but also ensures interpretable and differentiable models, which are fast to evaluate.

Sparsity within the groups of variables of the models of T, D, L and I_{sp} was ensured by the derivation of a variation of the Lasso [Tibshirani, 1994] in a structured multi-task setting. It relates in this sense to the Sparse-group Lasso [Friedman et al., 2010]. However, here we assume that the group sparsity pattern is known for each task and we only promote sparsity within the groups. It also differs from Multi-task Lasso [Argyriou et al., 2008, Obozinski et al., 2006], as we do not assume that all tasks share exactly the same features. Given that polynomial models present highly correlated features in practice, consistent feature selection is obtained by applying the Bolasso algorithm [Bach, 2008]. Section 3 contains our results on this subject.

Finally, we propose in section 4 an additional regularization, which injects more information into the problem. This allows to circumvent the issues related to the high correlation between the groups of variables, leading to better identifiability of the hidden elements T, D, L and I_{sp}.

All these measures in conjunction allow to obtain predictions with sufficient accuracy for trajectory optimization purposes, which was verified empirically. The results of our experiments based on real data can be found in section 5.

2 STRUCTURED MULTI-TASK MODEL

2.1 Hidden functions parametric models

In order to ensure interpretability and scalability, we chose to work only with parametric models of the hidden functions T, D, L and I_{sp}.

As we wanted to use the available expert knowledge of flight mechanics, new features were designed for each of these four elements based on the usual dependencies found in the literature [Roux, 2005, Hull, 2007]. Coincidentally, each of these four functions depend on different sets of 3 features, which can be obtained from the initial inputs of our problem $(\boldsymbol{x}, \boldsymbol{u})$ through nonlinear

mappings of the form

$$\varphi_\ell : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^3, \tag{6}$$

where $\ell \in \{T, D, L, I_{sp}\}$. More precisely, using the notation from table 1, we have:

$$\begin{aligned} \varphi_T(\boldsymbol{x}, \boldsymbol{u}) &= \big(N1, \rho(h), M(h, V)\big), \\ \varphi_D(\boldsymbol{x}, \boldsymbol{u}) &= \big(q(h, V), \alpha, M(h, V)\big), \\ \varphi_L(\boldsymbol{x}, \boldsymbol{u}) &= \big(q(h, V), \alpha, M(h, V)\big), \\ \varphi_{I_{sp}}(\boldsymbol{x}, \boldsymbol{u}) &= \big(SAT(h), h, M(h, V)\big). \end{aligned} \tag{7}$$

The reader is referred to the supplementary material for further details on the expressions of these common flight mechanics dependencies.

Furthermore, flight mechanics models found in the literature were mostly polynomials on the features obtained through (6). More precisely, many models found consisted in the product between one feature from the triplets listed in (7) and a linear combination of some monomials of the two remaining features. This led us to define the additional feature mapping:

$$\begin{array}{rccc} & \mathbb{R}^3 & \to & \mathbb{R}^r \\ \Phi_d : (z_1, z_2, z_3) & \mapsto & \left(z_1 \left(z_2^k z_3^{j-k} \right) ; \begin{matrix} j = 0, \dots, d \\ k = 0, \dots, j \end{matrix} \right), \end{array} \tag{8}$$

with $d \in \mathbb{N}^*$ and $r = \begin{pmatrix} d+2 \\ 2 \end{pmatrix}$. By picking $d > 1$, this last transformation allows to project our initial features into a higher dimensional space, potentially increasing the explanatory power of our model. Hence, for some $d_T, d_D, d_L, d_{Isp} > 1$, we assume the following linear structures:

$$\begin{aligned} T(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_T) &= X_T \cdot \boldsymbol{\theta}_T, \\ D(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_D) &= X_D \cdot \boldsymbol{\theta}_D, \\ L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_L) &= X_L \cdot \boldsymbol{\theta}_L, \\ I_{sp}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}_{Isp}) &= X_{Isp} \cdot \boldsymbol{\theta}_{Isp}, \end{aligned} \tag{9}$$

where

$$X_\ell = \Phi_{d_\ell}(\varphi_\ell(\boldsymbol{x}, \boldsymbol{u})) \in \mathbb{R}^{r_\ell}, \tag{10}$$

for $\ell \in \{T, D, L, I_{sp}\}$, and $\boldsymbol{\theta}_T, \boldsymbol{\theta}_D, \boldsymbol{\theta}_L, \boldsymbol{\theta}_{Isp}$ denote vectors of parameters. Note that, as Φ_d and φ_ℓ are differentiable for all $d \in \mathbb{N}$, it is also the case for models (9) as required.

2.2 Multi-task regression framework

In order to cast our identification problem into a regression problem, we first rewrite equations (3)-(5) as follows, to isolate the functions to be learned in the right-hand side of the system:

$$\begin{cases} m\dot{V} + mg\sin\gamma = T(\boldsymbol{x}, \boldsymbol{u})\cos\alpha - D(\boldsymbol{x}, \boldsymbol{u}), & (11) \\ mV\dot{\gamma} + mg\cos\gamma = T(\boldsymbol{x}, \boldsymbol{u})\sin\alpha + L(\boldsymbol{x}, \boldsymbol{u}), & (12) \\ 0 = T(\boldsymbol{x}, \boldsymbol{u}) + \dot{m} I_{sp}(\boldsymbol{x}, \boldsymbol{u}). & (13) \end{cases}$$

This led us to define the following set of regression models, after injecting T, D, L and I_{sp} linear structures previously derived in (9):

$$\begin{cases} Y_1 = X_{T1} \cdot \boldsymbol{\theta}_T - X_D \cdot \boldsymbol{\theta}_D + \varepsilon_1, & (14) \\ Y_2 = X_{T2} \cdot \boldsymbol{\theta}_T + X_L \cdot \boldsymbol{\theta}_L + \varepsilon_2, & (15) \\ 0 = X_T \cdot \boldsymbol{\theta}_T + X_{Ispm} \cdot \boldsymbol{\theta}_{Isp} + \varepsilon_3, & (16) \end{cases}$$

where $\varepsilon_1, \varepsilon_2, \varepsilon_3$, are random errors of mean 0 and

$$\begin{gathered} Y_1 = m\dot{V} + mg\sin\gamma, \quad Y_2 = mV\dot{\gamma} + mg\cos\gamma, \\ X_{T1} = X_T\cos\alpha, \\ X_{T2} = X_T\sin\alpha, \qquad X_{Ispm} = \dot{m}X_{Isp}. \end{gathered} \tag{17}$$

As commonly experienced in polynomial regression, the groups of interaction features $X_{T1}, X_{T2}, X_T, X_D, X_L$ and X_{Ispm} are highly correlated in practice. This causes numerical difficulties in solving regression problems (14)-(16), the parameter vectors $\boldsymbol{\theta}_T, \boldsymbol{\theta}_D, \boldsymbol{\theta}_L$ and $\boldsymbol{\theta}_{Isp}$ being able to compensate for each other. In addition, we want to avoid the trivial solution consisting in $\hat{\boldsymbol{\theta}}_T = \hat{\boldsymbol{\theta}}_{Ispm} = 0$ for problem (16) treated alone. Finally, the approach consisting in solving the three problems separately (e.g. through Ordinary Least Squares) would lead to three different estimators of T, inducing dynamics which are inconsistent with the known physics.

Because of the facts cited above, we chose to solve the three problems jointly in a multi-task regression framework:

$$Y = X\boldsymbol{\theta} + \varepsilon, \tag{18}$$

where $Y = (Y_1, Y_2, 0) \in \mathbb{R}^3$, $\varepsilon = (\varepsilon_1, \varepsilon_2, \varepsilon_3) \in \mathbb{R}^3$, $\boldsymbol{\theta} = (\boldsymbol{\theta}_T, \boldsymbol{\theta}_D, \boldsymbol{\theta}_L, \boldsymbol{\theta}_{Isp}) \in \mathbb{R}^p$ and X is a linear operator represented by the following sparse random matrix:

$$X = \begin{pmatrix} X_{T1}^\top & -X_D^\top & 0 & 0 \\ X_{T2}^\top & 0 & X_L^\top & 0 \\ X_T^\top & 0 & 0 & X_{Ispm}^\top \end{pmatrix} \in \mathbb{R}^{3\times p}. \tag{19}$$

By doing so, we enforced all three components of the final estimator to share the same $\hat{T}$ function, which complies with the dynamics structure (2)-(5). We also hoped to leverage the resulting coupling between tasks to increase the predictive accuracy, as observed in many other multi-task learning applications [Caruana, 1997, Evgeniou et al., 2005]. Furthermore, such framework helps to identify parameters $\boldsymbol{\theta}_T, \boldsymbol{\theta}_D, \boldsymbol{\theta}_L, \boldsymbol{\theta}_{Isp}$ in a high correlations setting, as explained with more details in subsection 4.

3 BLOCK SPARSE ESTIMATORS

3.1 Block sparse Lasso

We consider in this section that we have access to N random observations of $\boldsymbol{x}, \boldsymbol{u}, \dot{\boldsymbol{x}}$, allowing to build through the feature mappings (10), (17) and (19) a training set $\{(X^i, Y^i)\}_{i=1}^N$, where the X^i's are sampled from the distribution of

the random matrix (19). We assume this sample to be i.i.d and that all these random variables are centered and scaled to 1-standard deviation, except for the third component of the targets Y_3^i, which is assumed to be sampled from δ_0 a Dirac distribution in 0. In this context, the empirical risk minimization for the squared loss writes:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|Y^i - X^i\boldsymbol{\theta}\|_2^2, \tag{20}$$

which is equivalent to

$$\begin{aligned} \min_{\boldsymbol{\theta}_T,\boldsymbol{\theta}_D,\boldsymbol{\theta}_L,\boldsymbol{\theta}_{Isp}} & \|\boldsymbol{Y}_1 - \boldsymbol{X}_{T1}\boldsymbol{\theta}_T + \boldsymbol{X}_D\boldsymbol{\theta}_D\|_2^2 \\ & + \|\boldsymbol{Y}_2 - \boldsymbol{X}_{T2}\boldsymbol{\theta}_T - \boldsymbol{X}_L\boldsymbol{\theta}_L\|_2^2 \\ & + \|\boldsymbol{X}_T\boldsymbol{\theta}_T + \boldsymbol{X}_{Ispm}\boldsymbol{\theta}_{Isp}\|_2^2, \end{aligned} \tag{21}$$

where $\boldsymbol{X}_\ell$, $\ell \in \{T1, T2, T, D, L, I_{spm}\}$, (resp. $\boldsymbol{Y}_1, \boldsymbol{Y}_2$) denote matrices whose N rows correspond to the observations of X_ℓ (resp. observations of Y_1, Y_2).

Considering that the models from the literature used as inspiration for building our linear structures (9) were often composed of only a few monomials, feature selection seemed necessary here to try to avoid overfitting with an excessively complex model. Thus, the addition of an L^1-penalization to problem (20) was considered:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|Y^i - X^i\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1, \qquad \lambda > 0. \tag{22}$$

Problem (22) can be seen as a structured multi-task version of the Lasso [Tibshirani, 1994], called herein the Block sparse Lasso, where the residuals are in $\mathbb{R}^3$ and the features vector has been replaced by the sparse features matrix (19). Despite this differences, we can solve (22) with all the available algorithms for solving the Lasso, by noticing that the three terms forming the criterion from (21) can be written as a single-task least-squares criterion

$$\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\theta}\|_2^2, \tag{23}$$

where the targets observations have been stacked $\boldsymbol{Y} = (Y_1^1, \ldots, Y_1^N, Y_2^1, \ldots Y_2^N, 0, \ldots, 0)$ and the features observations have been used to build the block-sparse design matrix which gave the method its name:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_{T1} & -\boldsymbol{X}_D & 0 & 0 \\ \boldsymbol{X}_{T2} & 0 & \boldsymbol{X}_L & 0 \\ \boldsymbol{X}_T & 0 & 0 & \boldsymbol{X}_{Ispm} \end{pmatrix} \in \mathbb{R}^{3N \times p}. \tag{24}$$

Notice that, although $\boldsymbol{X}$ has a structure similar to the random matrix X from (19), it is not random and it has $3N$ rows instead of 3.

3.2 Bootstrap implementation

As it is the case for the Lasso (see e.g. [Zou, 2006]), it happens in practice that problem (22) leads to unstable solutions in a strong correlations setting. Hence,

we propose to use resampling in order to ensure consistent feature selection, as presented in [Bach, 2008].

We make the following standard assumptions on the joint distribution $P_{X,Y}$ of (X, Y):

(A1) The cumulant generating functions $\mathbb{E}\left[\exp(s\|X\|_2^2)\right]$ and $\mathbb{E}\left[\exp(s\|Y\|_2^2)\right]$ are finite for some $s > 0$.

(A2) The joint matrix of second order moments $Q = \mathbb{E}\left[XX^\top\right] \in \mathbb{R}^{p\times p}$ is invertible.

(A3) $\mathbb{E}[Y|X] = X \cdot \boldsymbol{\theta}$ and $\mathbb{V}[Y|X] = \sigma^2$ a.s. for some $\boldsymbol{\theta} \in \mathbb{R}^p$ and $\sigma \in \mathbb{R}_+^*$.

In our setting, assumption **(A1)** is equivalent to requiring finite cumulant generating functions for Y_1, Y_2, X_ℓ, $\ell \in \{T1, T2, T, D, L, I_{spm}\}$. This is the case when $\varepsilon_1, \varepsilon_2, X_\ell$, have compact support, which is verified in our case due to the physical nature of our data. Assumption **(A2)** summarizes here to each non-zero element of X having invertible second order moments, which has to be verified for our model to be identifiable. Assumption **(A3)** is of course equivalent to assuming the regression model (18). We denote by $J^* = \{j, \boldsymbol{\theta}_j \neq 0\}$ the sparsity pattern of $\boldsymbol{\theta}$.

The feature selection procedure proposed for our problem is described in algorithm 1.

Algorithm 1 Block sparse Bolasso

Require: training data $\mathcal{T} = \{(X^i, Y^i)\}_{i=1}^N$,
number of bootstrap replicates m,
L^1 penalization parameter λ,
for $k = 1$ **to** m **do**
  Generate bootstrap sample $\mathcal{T}^k$,
  Compute Block sparse Lasso estimate $\hat{\boldsymbol{\theta}}^k$ from $\mathcal{T}^k$,
  Compute support $J_k = \{j, \hat{\boldsymbol{\theta}}_j^k \neq 0\}$,
end for
Compute $J = \bigcap_{k=1}^m J_k$,
Compute $\hat{\boldsymbol{\theta}}_J$ from $\mathcal{T}_J = \{(X_J^i, Y^i)\}_{i=1}^N$ using (20).

In the previous algorithm, X_J^i denotes the matrix formed by the columns of X^i indexed by J.

As the Block sparse Lasso is equivalent to the Lasso with observations stacked in a certain way, the Bolasso consistency results summarized in Theorem 3.1 and proven in [Bach, 2008], also apply to algorithm 1:

Theorem 3.1 (Bolasso consistency) *Assume **(A1-3)** and $\lambda = \lambda_0 N^{-\frac{1}{2}}$, $\lambda_0 > 0$. Then the probability that the Bolasso does not exactly select the correct model, i.e., for all $m > 0$, $\mathbb{P}[J \neq J^*]$ has the following upper bound:*

$$\mathbb{P}[J \neq J^*] \leq mA_1 e^{-A_2 N} + A_3 \frac{\log N}{N^{1/2}} + A_4 \frac{\log m}{m}, \tag{25}$$

where $A_1, A_2, A_3, A_4 > 0$.

The main drawback of this general approach seems to be that it is based on solving a potentially large number of Block sparse Lasso problems, which makes it computationally costly. This flaw may be mitigated by the use of

existing efficient algorithms to solve the standard Lasso problem, such as the modified Least Angle Regression algorithm [Efron et al., 2004], whose computational complexity is equivalent to a matrix inversion.

4 REGULARIZATION FOR ENHANCED IDENTIFIABILITY

With the parameters $\hat{\boldsymbol{\theta}}$ obtained by solving any of the optimization problems (20) and (22), one can use (9) to have access to predictions of the hidden functions $\hat{T}, \hat{D}, \hat{L}$ and $\hat{I}_{sp}$. In practice, least-squares solutions produced by (20) lead to predicted thrust and specific impulse $\hat{T}, \hat{I}_{sp}$ really small compared to the known order of magnitude of these physical quantities. Moreover, when using sparsity promoting formulations such as the Block sparse Lasso (22) or Bolasso, $\hat{T}$ and $\hat{I}_{sp}$ are systematically equal to 0, all their parameters being rejected in the feature selection.

This seems to occur because we are regressing a function constantly equal to 0 in the last task (16), whose trivial solution is indeed setting all parameters of T and I_{sp} models to 0. This seems to be allowed by our model because the group of features of T are highly correlated to the features of D and L. Hence, it is possible to find solutions to problems (20) and (22) where targets Y_1 and Y_2 are completely explained by D and L, their parameters having compensated the absence of T.

To avoid this behavior, we introduce an additional regularization to our model. For this, we assume the availability of some prior estimator for any one of the hidden functions T, D, L, I_{sp}. This can be reduced to a constant estimator, equal to the known order of magnitude of given quantity. Without loss of generality, we suppose in what follows that $\tilde{I}_{sp}$ is such a prior estimator of I_{sp} and we define the regularized Block sparse Lasso problem:

$$\begin{aligned}\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|Y^i - X^i\boldsymbol{\theta}\|_2^2 + \lambda_2\|\tilde{I}_{sp}^i - X_{Isp}^i \cdot \boldsymbol{\theta}_{Isp}\|_2^2 \\ +\lambda_1\|\boldsymbol{\theta}\|_1, \qquad \lambda_1, \lambda_2 > 0,\end{aligned} \tag{26}$$

where, for $i = 1, \ldots, N$, we denote $\tilde{I}_{sp}^i = \tilde{I}_{sp}(\boldsymbol{x}^i, \boldsymbol{u}^i)$ the prediction of the prior estimator for some sample observations of states and controls.

Up to a scaling factor λ_2, the additional L^2 regularization term in (26) can be interpreted as if we had added one more task to system (14)-(16):

$$\tilde{I}_{sp} = X_{Isp} \cdot \boldsymbol{\theta}_{Isp} + \varepsilon_4, \tag{27}$$

where ε_4 is a random error with 0 mean. Hence, algorithm 1 can be run with the regularized Block sparse Lasso (26) instead of (22) by replacing X^i, Y^i by

$$\tilde{X}^i = \begin{pmatrix} (X_{T1}^i)^\top & -(X_D^i)^\top & 0 & 0 \\ (X_{T2}^i)^\top & 0 & (X_L^i)^\top & 0 \\ (X_T^i)^\top & 0 & 0 & (X_{Ispm}^i)^\top \\ 0 & 0 & 0 & \lambda_2(X_{Isp}^i)^\top \end{pmatrix}, \tag{28}$$

and $\tilde{Y}^i = (Y_1^i, Y_2^i, 0, \lambda_2\tilde{I}_{sp}^i)$.

Empirical results presented in section 5 show that predictions of the hidden functions with the correct orders of magnitude can be obtained with this technique.

5 EXPERIMENTS

5.1 Data set description and preprocessing

QAR data description We present in this section numerical experiments carried using real flight data, extracted from Quick Access Recorder devices (QAR) of 25 medium haul aircraft of the same type. Although the original data set contains raw measurements of thousands of different variables, only the following were used herein: $(h, M, C, N_1, SAT, \beta)$. The reader is referred to table 1 for the notations. Only data concerning the climb phase of the flights were kept, i.e. data corresponding to altitudes between FL50 = 5 000 ft and the *top of climb* (cruise altitude), specific to each flight. This was justified by the fact that the estimated dynamics would be used for the optimization of the climb profiles of these aircraft. The obtained data set contains 8 261 619 observations, made of 10 471 different flights sampled at 1 measurement per second.

Derivation of state and control observations The raw data was smoothed using univariate smoothing splines and then used to derive all the state and control variables $\boldsymbol{x}, \boldsymbol{u}$ presented in section 1.1. This was done by the means of standard flight mechanics formulas, which are listed in the supplementary material. Observations of state derivatives $\dot{\boldsymbol{x}}$ were then computed analytically, based on the previous formulas and on the splines-derivatives of the smoothed measured variables. The features used to build the linear models of T, D, L and I_{sp} were obtained using the feature mapping (10), where $d_T = 4$ and $d_D = d_L = d_{Isp} = 3$.

5.2 Experiments design

Feature selection In order to assess the quality of the feature selection performed by the Block sparse Bolasso presented in section 3.2, algorithm 1 was run seperately on the datasets of each aircraft. The sparsity patterns obtained were then compared. The regularization parameter λ_2 introduced in section 4 was set to 200, which is justified in section 5.3. The prior model $\tilde{I}_{sp}$ used here to define such regularization was found in [Roux, 2005]. The L^1 penalty parameter λ_1 of the regularized Block sparse Lasso (26) was set using 30-fold cross-validation based on the squared-loss. This was performed separately for each dataset, on 33%-validation sets, and done a single time, prior to the m replications of the Bolasso. For all experiments, the number of bootstrap replications was set to $m = 128$. Solving the mutiple Block sparse Lasso problems was done using the Least Angle Regression algorithm, implemented in Python's SCIKIT-LEARN.LINEAR_MODEL library [Pedregosa et al., 2011].

Quality of the state equation estimations The quality of the estimators obtained using the Block sparse Bolasso algorithm was assessed using a subset

of $q = 424$ flights, corresponding to a single aircraft and comprising 334 531 observations. Our model was trained on $q-1$ flights, leaving out one flight (chosen randomly) for testing. The L^1 parameter was selected using cross validation, prior to the execution of algorithm 1, as previously explained for the feature selection assessment. The calibration of λ_2 was done by solving (26) with varying parameter values. We kept the parameter leading to predicted $\hat{T}, \hat{D}, \hat{L}$ and $\hat{I}_{sp}$ agreeing with the physical context.

Let N_{test} denotes the number of observations from the test flight. As a first quality criterion, state derivatives of the test flight are predicted using our trained model and are then compared to the observed states derivatives of given flight. Considering that the goal of our approach is to deliver models that are suitable for trajectory optimization, another quality criterion was considered, based on comparing the observed state and control variables of the test flight $\{\boldsymbol{x}_m^i, \boldsymbol{u}_m^i\}_{i=1}^{N_{test}}$ to the solution of the following trajectory optimization problem:

$$\begin{aligned} \min_{\boldsymbol{u}\in\mathbb{U}} \int_0^{t_f} \left(\|\boldsymbol{u}(t) - \boldsymbol{u}_m(t)\|_{\boldsymbol{u}}^2 + \|\boldsymbol{x} - \boldsymbol{x}_m(t)\|_{\boldsymbol{x}}^2 \right) dt \\ \text{s.t.} \quad \dot{\boldsymbol{x}} = \hat{g}(\boldsymbol{x}, \boldsymbol{u}), \end{aligned} \tag{29}$$

where t_f denotes the time horizon of the test flight and $\|\cdot\|_{\boldsymbol{u}}$, $\|\cdot\|_{\boldsymbol{x}}$ denote scaling norms. Note that the aircraft dynamics in this optimal control problem (29) were defined with the trained state equation model. Solutions were obtained using the software BOCOP [Bonnans et al., 2017].

5.3 Results

The results of the feature selection performed on the 25 data sets are on figure 1. Each row of these matrices corresponds to a different aircraft and each column to a different feature. The cells colors encode the frequency of selection of given feature for given aircraft across all the Block sparse Lasso executions. It can be observed that most dark columns are quite homogeneous, which indicates that similar features were selected for the majority of aircraft. This seems to validate our approach for this kind of data, as one would expect that airplanes of the same type should have physical models for the thrust, drag, lift and specific impulse with similar structures. As an example, a common sparse model for the thrust force T would be here made of the features: $X_T = N_1(1, \rho M^2, M^4, \rho^2 M^2, \rho^3 M, \rho^4)$.

Figure 2 illustrates the effect of the regularization presented in section 4 on the test flight predictions obtained with our estimator. Left frames show that it affects mainly $\hat{T}$ and $\hat{I}_{sp}$, which mirror each other because of equation (16). The drag D seems to vary slightly when the value of λ_2 is changed, while L is unchanged. It is well-known that during climb, the thrust force of commercial aircraft decreases. However, it may be observed that for a small value of $\lambda_2 = 20$, $\hat{T}$ and $\hat{I}_{sp}$ are flat, which seems to indicate that more regularization is needed. A high value of $\lambda_2 = 2000$ brought our predicted thrust to increase quickly at the end of the climb, which does not seem correct either. We seem to have constrained excessively our model of $\hat{I}_{sp}$. Hence, $\lambda_2 = 200$ seems like the best choice here. Right frames of figure 2 show that the regularization parameter λ_2 has little effect on the final predictions of the state function estimator $\hat{g}$. These predictions seem to be relatively good approximations of the observed states

derivatives for our purposes. This can be seen for instance in figure 3, which shows the obtained solution for problem (29), illustrating that it is possible to resimulate the test flight using our estimator.

6 CONCLUSIONS

We developed in this paper a new method for aircraft dynamics identification based on a multi-task learning framework. As it has been designed for trajectory optimization purposes, it delivers interpretable models, which agree with the flight mechanics common knowledge and can be quickly evaluated and differentiated by an optimization solver. Our estimator is able to obtain consistent and structured sparsity, thanks to an original regression formulation and the use of a previously suggested bootstrap strategy: the Bolasso. We also showed that such estimator could be trained using available, well-known and efficient algorithms. The numerical experiments that we carried using real data from 25 different aircraft strongly suggest that our approach performs reliable feature selection. Furthermore, although this was not our main design criterion, satisfactory accuracy was evidenced by our results, at the cost of less than 10 minutes training using a laptop and a data set of 334 531 observations. The main drawback of our approach seems to be the need of a prior estimator of one of the hidden functions.

An interesting question raised by this approach is whether the trajectories optimized using these types of estimators lie close to the observed flights used to compute the model. This could give an indication on the reliability of such trajectories, which is currently under study. Moreover, considering the many issues caused by the strong correlations among features of T, D, L and I_{sp}, the use of different models (e.g. using orthogonal polynomials) should also be investigated.

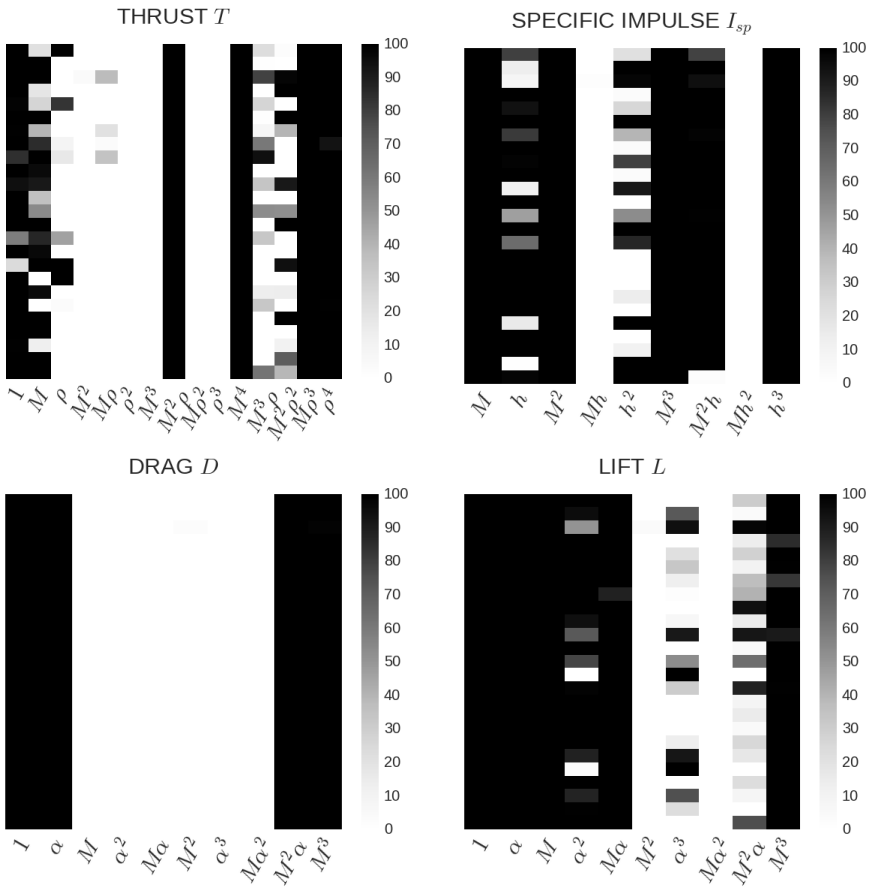

Figure 1: Feature selection results for the thrust, drag, lift and specific impulse models. The columns correspond to possible features for T/N_1, D/q, L/q and I_{sp}/SAT.

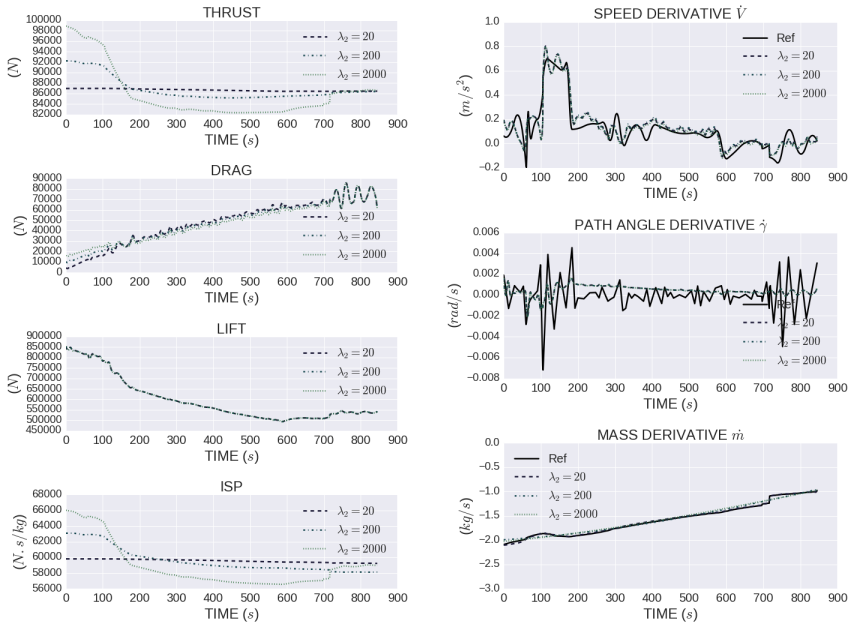

Figure 2: (*Left*) Thrust, Drag, Lift and Specific impulse predicted for the test flight with different regularization parameters λ_2. (*Right*) Respective states derivatives predictions.

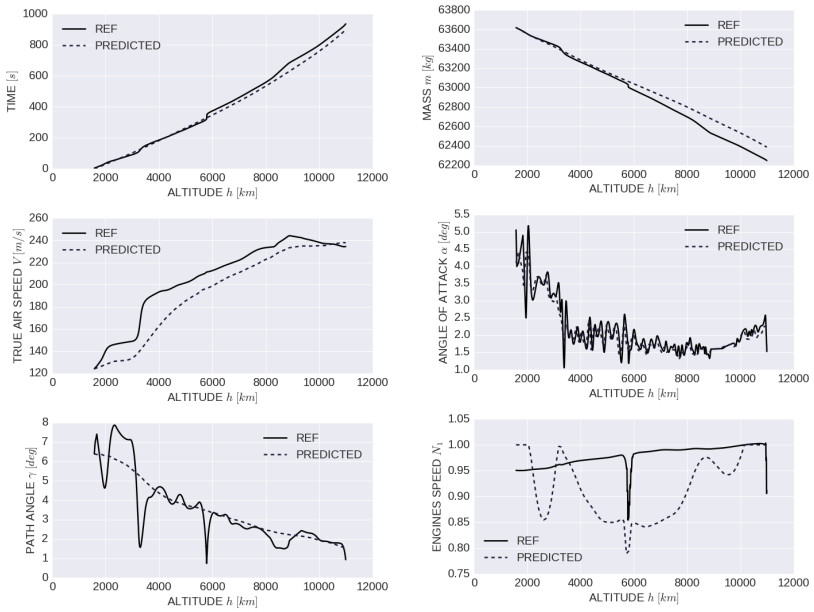

Figure 3: Resimulated trajectory obtained by solving problem (29) with BOCOP.

References

[Argyriou et al., 2008] Argyriou, A., Evgeniou, T., and Pontil, M. (2008). Convex multi-task feature learning. *Machine Learning Journal*.

[Bach, 2008] Bach, F. (2008). Bolasso: model consistent Lasso estimation through the bootstrap. In *Proceedings of the 25th international conference on Machine learning*, pages 33–40.

[Bonnans et al., 2017] Bonnans, J. F., Giorgi, D., Grelard, V., Heymann, B., Maindrault, S., Martinon, P., Tissot, O., and Liu, J. (2017). Bocop – A collection of examples. Technical report, INRIA.

[Caruana, 1997] Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1):41–75.

[Efron et al., 2004] Efron, B. et al. (2004). Least angle regression. *The Annals of Statistics*, 32:407–499.

[Evgeniou et al., 2005] Evgeniou, T., Micchelli, C. A., and Pontil, M. (2005). Learning multiple tasks with kernel methods. *Journal Machine Learning Research.*, 6:615–637.

[Friedman et al., 2010] Friedman, J., Hastie, T., and Tibshirani, R. (2010). A note on the Group Lasso and a Sparse group Lasso. arXiv:1001.0736.

[Hull, 2007] Hull, D. G. (2007). *Fundamentals of Airplane Flight Mechanics*. Springer.

[Jategaonkar, 2006] Jategaonkar, R. V. (2006). *Flight Vehicle System Identification: A Time Domain Methdology*. AIAA.

[Klein and Morelli, 2006] Klein, V. and Morelli, E. A. (2006). *Aircraft System Identification*. AIAA.

[Maine and Iliff, 1985] Maine, R. E. and Iliff, K. W. (1985). *Identification of Dynamic Systems: Theory and Formulation*. NASA, STIB.

[Maine and Iliff, 1986] Maine, R. E. and Iliff, K. W. (1986). *Application of Parameter Estimation to Aircraft Stability and Control: The Output-error Approach*. NASA, STIB.

[Morelli, 2006] Morelli, E. A. (2006). Practical aspects of the equation-error method for aircraft parameter estimation. In *AIAA Atmospheric Flight Mechanics Conference*. AIAA.

[Obozinski et al., 2006] Obozinski, G., Taskar, B., and Jordan, M. I. (2006). Multi-task feature selection. In *ICML-06 Workshop on Structural Knowledge Transfer for Machine Learning*.

[Pedregosa et al., 2011] Pedregosa, F. et al. (2011). Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830.

[Peyada and Ghosh, 2009] Peyada, N. K. and Ghosh, A. K. (2009). Aircraft parameter estimation using a new filtering technique based upon a neural network and Gauss-Newton method. *The Aeronautical Journal*, 113(1142):243–252.

[Peyada et al., 2008] Peyada, N. K., Sen, A., and Ghosh, A. K. (2008). Aerodynamic characterization of hansa-3 aircraft using equation error, maximum likelihood and filter error methods. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*.

[Rommel et al., 2017] Rommel, C., Bonnans, J. F., Gregorutti, B., and Martinon, P. (2017). Aircraft dynamics identification for optimal control. In *Proceedings of the 7th European Conference for Aeronautics and Aerospace Sciences*.

[Roux, 2005] Roux, E. (2005). *Pour une approche analytique de la dynamique du vol*. PhD thesis, SUPAERO.

[Tibshirani, 1994] Tibshirani, R. (1994). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society*, 58:267–288.

[Yuan and Lin, 2005] Yuan, M. and Lin, Y. (2005). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society*, 68:49–67.

[Zou, 2006] Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429.